We claim:

1. A computerized method for managing a directed labeled graph, comprising:
generating the directed labeled graph;
generating a cursor of the directed labeled graph; and
navigating through the directed labeled graph using the cursor.

2. A computerized method for managing a directed labeled graph as in claim 1, wherein the directed labeled graph is selected from the group consisting of conceptual schemas, metaschemas, and instance data;

3. A computerized method for managing a graph comprising:
generating a cursor of the graph; and
navigating through the graph using the cursor.

4. The computerized method for managing a graph, as in claim 3, further comprising
generating a constraint graph, the constraint graph having at least one constraint on the managing of the graph and wherein the generating a cursor further comprises the cursor being associated with the constraint graph.

5. The computerized method for managing a graph, as in claim 4, wherein the generating a constraint graph further comprises:
obtaining a handle to a required namespace node, yielding a namespace handle;
obtaining a handle to a variable using the namespace handle;
determining the identity of each of a plurality of nodes and links associated with the namespace handle;
obtaining a handle to each of the plurality of nodes and links using the namespace handle;

generating a segment; and

storing the segment.

6. The computerized method for managing a graph, as in claim 4, wherein the generating a segment further comprises:

create a second cursor, in which the variable is bound to one of the plurality of nodes and links.

7. The computerized method for managing a graph, as in claim 3, wherein the navigating through the graph using the cursor further comprises:

bind a variable to the first instance of the plurality of nodes and links;

bind a variable to the last instance of the plurality of nodes and links;

bind a variable to the next instance of the plurality of nodes and links; and

bind a variable to the previous instance of the plurality of nodes and links;

8. The computerized method for managing a graph, as in claim 3, further comprising:

adding a linked node to the graph;

detaching a linked node from the graph;

9. The computerized method for managing a graph, as in claim 3, further comprising:

retrieving information from the graph;

setting the value of a node that a specified variable is bound;

beginning an atomic transaction;

ending the atomic transaction;

committing the current atomic transaction;

aborting the current atomic transaction;

cloning the cursor;

entering a snapshot mode;

refreshing the snapshot mode;

locking a node;

locking a pattern;

specifying a level of authorization and access permission in levels of granularity;

mapping a link and a node to a data source;

specifying a variable to receive notification of change; and

specifying a pattern to receive notification of change.

10. A computerized system comprising:

a directed labeled graph having node and links;

an annotated schema runtime engine operably coupled to the directed labeled graph; and

an apparatus having at least one a variable bound to a node or link in the directed labeled graph operably coupled to the annotated schema runtime engine.

11. The computerized system as in claim 10, wherein the apparatus having at least one variable bound to a node or link in the directed labeled graph comprises:

an annotated schema runtime interface; and

an annotated schema runtime cursor interface.

12. The computerized system as in claim 11, further comprising a client operably coupled to the annotated schema runtime interface and operably coupled to the annotated schema runtime cursor interface.

13. The computerized system as in claim 11, wherein the annotated schema runtime engine, the annotated schema runtime interface, and the annotated schema runtime

cursor interface are objects.

14. The computerized system as in claim 13, wherein the annotated schema runtime cursor object comprises a plurality of ASR cursor objects.

15. The computerized system as in claim 14, wherein the plurality of ASR cursor objects are a plurality of ASR cursor COM objects.

16. A computer-readable medium having computer-executable instructions to a cause a computer to perform a method comprising:

generating a cursor of a graph;

navigating in the graph using the cursor; and

retrieving data from nodes.

17. A system managing graph data, comprising:

a processor;

a storage device coupled to the processor, having graph data stored thereon, the graph data selected from the group consisting of conceptual schemas, metaschemas, and instance data; and

software means operative on the processor for:

generating a cursor of the graph data; and

navigating through the graph data using the cursor.

18. A computer-readable medium having stored thereon a data structure comprising:

a graph having a plurality of nodes and links; and

at least one variable bound to one of the plurality of nodes and links.

19. The computer-readable medium having stored thereon a data structure as in claim 4, wherein:

the linked list further comprises a graph; and

the bound variable further comprises a graph cursor.

20. The computer-readable medium having stored thereon a data structure as in claim 19, wherein the bound variable is bound using a restriction.

21. The computer-readable medium having stored thereon a data structure as in claim 4, wherein the bound variable is associated with a constraint graph.

22. A computer-readable medium having stored thereon a data structure for referencing a subset of data in a directed label graph, comprising:

a plurality of legs identifying a selected subset of nodes in the graph; and

a cursor used to point to said selected subset of nodes.

23. A computer-readable medium having computer-executable instructions for implementing a spider system, comprising:

an IASR application program interface configured as a software module to interface with an annotated schema runtime engine and to handle requests from a client for a spider function; and

an IASR cursor application program interface configured as a software module to interface with an annotated schema runtime engine and to handle requests from a client for a spider function.

24. An IASR cursor application program interface for creating a cursor by creating a new node in a graph from information comprising a variable and a constraint.

25. The IASR cursor application program interface as in claim 24 further comprising an IASR cursor application program interface for storing new nodes and links.

26. The IASR cursor application program interface as in claim 24 further comprising an IASR cursor application program interface for finding the first node or link in the graph in reference to the constraint from information comprising a variable specifying the first node or link in the graph, with respect to the current sort order.

27. The IASR cursor application program interface as in claim 24 further comprising an IASR cursor application program interface for finding the last node or link in the graph in reference to the constraint from information comprising a variable specifying the last node or link in the graph, with respect to the current sort order.

28. The IASR cursor application program interface as in claim 24 further comprising an IASR cursor application program interface for finding the next node or link in the graph in reference to the constraint from information comprising a variable specifying the next node or link in the graph, with respect to the current sort order.

29. The IASR cursor application program interface as in claim 24 further comprising an IASR cursor application program interface for finding the previous node or link in the graph in reference to the constraint from information comprising a variable specifying the previous node or link in the graph, with respect to the current sort order.

30. A data structure stored on a computer-readable medium for a schema comprising:
a field storing data representing an entity;
a field storing data representing a relation; and
a field storing data representing a value.

31. The data structure as in claim 30, wherein the field storing data representing an entity and

wherein the field storing data representing a relation provides a context that associates

the field storing data representing and entity and the field storing data representing a value in a meaningful fashion.